Claims

[c1] 1. A system for distributed processing, comprising at least one Distributed Working Environment for Application Processing (DWEAP), wherein the at one DWEAP is coupleable to an Internet Service Provider (ISP), wherein the ISP is coupled to an Internet, wherein a main server is coupled to the Internet, wherein an application is coupled to the main server, and wherein each said DWEAP has an application cache for caching a runnable module (RUM) of the application.

[c2] 2. The system of claim 1, wherein a database is coupled to the main server, wherein a database cache is coupled to at least one of said DWEAPs for caching a portion of the database, and wherein the portion of the database is required for execution of the application.

[c3] 3. The system of claim 1, wherein the at least one DWEAP includes a first DWEAP and a second DWEAP, wherein the application cache of the first includes a first RUM of the application, wherein the application cache of the second DWEAP includes a second RUM of the application, and wherein execution of the application includes execution of the first RUM on the first DWEAP and execution of the second RUM on the second DWEAP.

[c4] 4. The system of claim 3, further comprising a database coupled to the main server, wherein a first portion of the database is cached on the first DWEAP, wherein a second portion of the database is cached on the second DWEAP, wherein execution of the first RUM includes utilization of the first portion of the database, and wherein execution of the second RUM includes utilization the second portion of the database.

[c5] 5. The system of claim 1, further comprising a client coupled to a first of the at least one DWEAP.

[c6] 6. The system of claim 5, wherein the application is executable such that a portion of the application is runnable closer to the client.

[c7] 7. The system of claim 6, wherein the portion of the application is the entire

application.

[c8] 8. The system of claim 5, wherein the application is executable such that a portion of the application is runnable near the client.

[c9] 9. The system of claim 5, wherein the application is executable such that a portion of the application is runnable on the client.

[c10] 10. The system of claim 5, wherein the application is executable such that a first portion of the application is runnable closer to the client and a second portion of the application is runnable on the main server.

[c11] 11. The system of claim 5, wherein the application is executable such that a portion of the application is runnable closer to the client with a lower latency for execution of the application than if the entire application were run on the main server.

[c12] 12. The system of claim 1, wherein the at least one DWEAP is coupled to the ISP.

[c13] 13. A system for distributed processing, comprising at least one node, wherein the at least one node is coupleable to an Internet, wherein a main server is coupled to the Internet, wherein an application is coupled to the server, wherein each said node comprises a network server, a Distributed Working Environment for Application Processing (DWEAP) for execution of a runnable module (RUM) of the application, and an application module cache for caching the RUM.

[c14] 14. The system of claim 13, wherein a database is coupled to the main wherein at least one of said nodes further comprises a synchronized cache for caching a portion of the database, and wherein the portion of database is required for execution of the application.

[c15] 15. The system of claim 13, wherein the at least one node includes a first node and a second node, wherein the application cache of the first node includes a first RUM of the application, wherein the application cache of the

second node includes a second RUM of the application, and wherein an execution of the application includes execution of the first RUM on the of the first node and execution of the second RUM on the DWEAP of the second node.

[c16] 16. The system of claim 15, further comprising a database coupled to the main server, wherein a first portion of the database is cached on the first node, wherein a second portion of the database is cached on the second wherein execution of the first RUM includes utilization of the first portion of the database, and wherein execution of the second RUM includes utilization the second portion of the database.

[c17] 17. The system of claim 15, wherein the first node is coupled to the second node.

[c18] 18. The system of claim 13, further comprising a client coupled to a first of the at least one node.

[c19] 19. The system of claim 18, further comprising a DWEAP coupled to the

[c20] 20. The system of claim 18, further comprising an Internet Service Provider (ISP) coupled to the first node and to the client.

[c21] 21. The system of claim 18, wherein the application is executable such that a portion of the application is runnable closer to the client.

[c22] 22. The system of claim 18, wherein the portion of the application is the application.

[c23] 23. The system of claim 18, wherein the application is executable such that a portion of the application is runnable near the client.

[c24] 24. The system of claim 18, wherein the application is executable such that a portion of the application is runnable on the client.

[c25] 25. The system of claim 18, wherein the application is executable such that a first portion of the application is runnable closer to the client and a second

portion of the application is runnable on the main server.

[c26] 26. The system of claim 18, wherein the application is executable such that a portion of the application is runnable closer to the client with a lower latency for execution of the application than if the entire application were run on the main server.

[c27] 27. The system of claim 13, wherein the at least one node is coupled to the Internet.

[c28] 28. An application comprising a plurality of runnable modules (RUMs), the application is coupled to a main server, wherein the main server is to an Internet, wherein at least one RUM of the plurality of RUMs is cached at one or more locations, and wherein the application is executable such that at least one RUM is runnable closer to a client.

[c29] 29. The application of claim 28, wherein each of the one or more locations includes a Distributed Working Environment for Application Processing (DWEAP), and wherein the at least one RUM is runnable on at least one of the DWEAPs.

[c30] 30. The application of claim 29, wherein each of the one or more locations include a node of a system for distributed processing, and wherein the node at each location includes the DWEAP at each location.

[c31] 31. The application of claim 29, wherein the client is coupled to a DWEAP of the DWEAPS at the one or more locations, and wherein the application is executable such that the at least one RUM is runnable on the client.

[c32] 32. The application of claim 28, wherein the application is executable such that the at least one RUM is runnable near the client.

[c33] 33. The application of claim 28, wherein a non-cached RUM of the plurality RUMS is runnable only on the main server.

[c34] 34. The application of claim 28, wherein a first RUM of the at least one RUM

cached at a first DWEAP within the one or more locations, and wherein a second RUM of the at least one RUM is cached at a second DWEAP within the one or more locations.

[c35] 35. The application of claim 34, wherein the first RUM is runnable on the client, and wherein the second RUM is runnable near the client.

[c36] 36. The application of claim 34, wherein the first RUM is runnable near the client, and wherein the second RUM is runnable near the client.

[c37] 37. The application of claim 34, wherein a non-cached RUM of the plurality RUMS is runnable only on the main server.

[c38] 38. The application of claim 28, wherein execution of the application that includes running the at least one RUM closer to the client has a lower latency than if the entire application were run on the main server.

[c39] 39. A database coupled to a main server, wherein the main server is coupled to an Internet, wherein the database comprises a plurality of database portions, wherein at least one database portion of the plurality of database portions is cached at one or more locations, and wherein the one or more locations are closer to a client.

[c40] 40. The database of claim 39, wherein each of the one or more locations includes a Distributed Working Environment for Application Processing (DWEAP) for running a runnable module (RUM) which utilizes whatever of the database is cached at the DWEAP.

[c41] 41. The database of claim 40, wherein each of the one or more locations include a node of a system for distributed processing, and wherein the node at each location includes the DWEAP at each location.

[c42] 42. The database of claim 40, wherein the client is coupled to a DWEAP of DWEAPS at the one or more locations.

[c43] 43. The database of claim 39, wherein the one or more locations are near the

client.

[c44] 44. The database of claim 39, wherein the database includes a non-cached portion that is not cached away from the main server.

[c45] 45. The database of claim 39, wherein a first database portion of the at least one database portion is cached at a first location of the one or more and wherein a second database portion of the at least one database portion cached at a second location of the one or more locations.

[c46] 46. The database of claim 45, wherein the first location is on the client, and wherein the second location is near the client.

[c47] 47. The database of claim 45, wherein the first location is near the client, wherein the second location is near the client.

[c48] 48. The database of claim 45, wherein the database includes a non-cached portion that is not cached away from the main server.

[c49] 49. A system of communication, said system comprising a channel architecture having at least one channel path, wherein each said channel comprises N channels denoted as $C_1, C_2, \ldots, C_{N-1}, C_N$ in a hierarchical structure of $/C_1 /C_2 /\ldots /C_{N-1} /C_N$, wherein N is at least 2, wherein C_1 a root channel, wherein C_n is a subchannel having C_{n-1} as a parent for n= 2,3,..., N, wherein C_n has an ontology O_m for m=1,2,...,N, and wherein O_i comprises all attributes included in O_{i-1} for i=2,3,...N.

[c50] 50. The system of claim 49, wherein the at least one channel path comprises first channel path P_1 and a second channel path P_2, and wherein P_1 and P have a same root channel.

[c51] 51. The system of claim 49, wherein the at least one channel path comprises first channel path P_1 and a second channel path P_2, and wherein P_1 and P have a different root channel.

[c52] 52. The system of claim 49, wherein the at least one channel path comprises

first channel path, wherein the first channel path has a subchannel C and a parent channel P of the subchannel C, and wherein the ontology of C at least one attribute not included in the ontology of P.

[c53] 53. The system of claim 49, wherein the at least one channel path comprises first channel path, wherein the first channel path has a subchannel C, and wherein C comprises a component.

[c54] 54. The system of claim 53, wherein the component includes a static object.

[c55] 55. The system of claim 54, wherein the static object is selected from the group consisting of a message, a downloadable file, and a graphics file.

[c56] 56. The system of claim 53, wherein the component includes a dynamic object.

[c57] 57. The system of claim 53, wherein the component includes an executable module.

[c58] 58. The system of claim 53, wherein C includes an executable interface for executing a runnable module (RUM).

[c59] 59. The system of claim 58, wherein the executable interface stores the RUM.

[c60] 60. The system of claim 58, wherein the executable interface does not store the RUM.

[c61] 61. The system of claim 58, wherein C comprises meta data, wherein the data includes the name of the RUM, and wherein the meta data points to a method to call within the RUM.

[c62] 62. The system of claim 53, wherein C comprises meta data.

[c63] 63. The system of claim 62, wherein the meta data includes Multi-purpose Internet Mail Extensions (MIME) types.

[c64] 64. The system of claim 63, wherein the MIME types include a type selected from the group consisting of a deliverable component, an edge component,

deliverable executable component, and a channel component.

[c65] 65. The system of claim 53, wherein the component includes a message.

[c66] 66. The system of claim 53, wherein the component includes an attribute selected from the group consisting of expiration time, quality of service, source, name, globally unique identifier, owner, access control list, and combinations thereof.

[c67] 67. The system of claim 49, wherein the at least one channel path comprises first channel path, wherein the first channel path has a subchannel C and a parent channel P of the subchannel C, wherein the ontology of P includes an attribute A, wherein the ontology of C includes the attribute A and an B, wherein a first message having the attributes A and B has been posted to wherein a subsequent message having the attribute A and not having the attribute B has been posted to P, and wherein the subsequent message was derived from the first message.

[c68] 68. The system of claim 49, wherein a specified channel of the at least one channel path is destroyable.

[c69] 69. The system of claim 49, wherein a subchannel of a current channel of the at least one channel path may be created.

[c70] 70. The system of claim 49, wherein a current channel of the at least one channel path may be subscribed to a specified channel of the at least one channel path.

[c71] 71. The system of claim 49, wherein a current channel of the at least one channel path may be unsubscribed from a specified channel of the at least channel path.

[c72] 72. A system of communication over a distributed processing network, said system comprising:
the distributed processing network having a plurality of nodes, wherein each node is coupleable to an Internet, wherein a main server is coupled to the

Internet, wherein an application is coupled to the main server, wherein each node comprises a network server, a Distributed Working Environment for Application Processing (DWEAP) for execution of a runnable module (RUM) of the application, and an application module cache for caching the RUM; and a channel architecture having at least one channel path, wherein each said channel path comprises N channels denoted as C_1, C_2,... , C_{N-1}, C_N in hierarchical structure of /C_1 /C_2 /... /C_{N-1} /C_N, wherein N is at least 2, wherein C_1 is a root channel, wherein C_n is a subchannel having C_{n-1} as a parent channel for n= 2,3,..., N, wherein C_n has an ontology O_m for m=1,2,...,N, and wherein O_i comprises all attributes included in O_{i-1} for i=2,3,...N, and wherein the distributed processing network uses the architecture to transfer the RUMs from the main server to the DWEAPs.

[c73] 73. The system of claim 72, wherein a database is coupled to the main wherein at least one of said nodes further comprises a synchronized cache for caching a portion of the database, wherein the portion of the database is required for execution of the application, and wherein the distributed processing network uses the channel architecture to transfer the portion of the database from the main server to the synchronized database cache of the at least one of said nodes.

[c74] 74. A subscription structure, comprising:
a first channel;
a second channel to which the first channel is subscribed; and
a component in the first channel, wherein the component is intended to be transported from the first channel to the second channel, wherein the component is intended to deliver at least one attribute to the second and wherein the first channel determines the at least one attribute.

[c75] 75. The subscription structure of claim 74, further comprising a subscription filter through which the component must pass while en route from the first channel to the second channel, wherein the subscription filter tests the at least one attribute against a conditional match expression.

[c76] 76. The subscription structure of claim 75, further comprising a channel through which the component must pass while en route from the first to the second channel, wherein the component must pass through the subscription filter before passing through the channel filter, and wherein the channel filter determines whether the component may be posted to the second channel.

[c77] 77. The subscription structure of claim 74, further comprising a channel through which the component must pass while en route from the first to the second channel, and wherein the channel filter determines whether component may be posted to the second channel.

[c78] 78. The subscription structure of claim 74, wherein the component is a message.

[c79] 79. The subscription structure of claim 78, wherein the second channel comprises a logical filter which acts upon the message.

[c80] 80. The subscription structure of claim 79, wherein the logical filter validates the message.

[c81] 81. The subscription structure of claim 79, wherein the logical filter refers to logical object.

[c82] 82. The subscription structure of claim 79, wherein the logical filter is an expression in terms of attributes of a third channel.

[c83] 83. The subscription structure of claim 74, wherein the first channel is an interface channel, wherein the second channel includes an executable component, and wherein upon arrival of the component into the second channel the component causes the executable component to call a function.

[c84] 84. The subscription structure of claim 74, wherein the first channel requires certificate-based authentication of the second channel prior to allowing the component to be transported from the first channel to the second channel.

[c85] 85. The subscription structure of claim 74, wherein the second channel requires certificate-based authentication of the first channel prior to the component to be received by second channel from the first channel.

[c86] 86. A two-way nodal communication network, comprising:
a first node;
a second node, wherein an object O may be sent from the first node to the second node or from the second node to the first node, wherein the first sends or receives O in a form F_1, and wherein the second node sends or receives O in a form F_2; and
a transport strategy, wherein if O is sent from the first node to the second node then the transport strategy converts O from F_1 into a common form after O is sent by the first node and the transport strategy subsequently converts O from the common form into F_2 prior to O being received by the second node, and wherein if O is sent from the second node to the first then the transport strategy converts O from F_2 into the common form after is sent by the second node and the transport strategy subsequently from the common form into F_1 prior to O being received by the first node.

[c87] 87. The two-way nodal communication network of claim 86, wherein F_1 differs from F_2.

[c88] 88. The two-way nodal communication network of claim 86, wherein $F_1 = F_2$.

[c89] 89. The two-way nodal communication network of claim 86, wherein O is a message.

[c90] 90. The two-way nodal communication network of claim 89, wherein F_1 is a first communication protocol, and wherein F_2 is a second communication protocol.

[c91] 91. The two-way nodal communication network of claim 86, wherein O is a runnable module.

[c92] 92. The two-way nodal communication network of claim 91, wherein F_1 is a function of a first software language, and wherein F_2 is a function of a software language.

[c93] 93. The two-way nodal communication network of claim 86, wherein F_1 is a function of a first operating system platform, and wherein F_2 is a function a second operating system platform.

[c94] 94. The two-way nodal communication network of claim 86, wherein F_1 is a function of a first hardware device platform, and wherein F_2 is a function second hardware device platform.

[c95] 95. The two-way nodal communication network of claim 86, wherein F_1 is a function of a first software language, and wherein F_2 is a function of a software language.